Author manuscript, published in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 1737-1746, 2011.
DOI: 10.1145/1978942.1979195

KronoMiner: Using Multi-Foci Navigation for the Visual Exploration of Time-Series Data

Jian Zhao[1] **Fanny Chevalier**[2] **Ravin Balakrishnan**[1]

[1]Department of Computer Science
University of Toronto
{jian|ravin}@dgp.toronto.edu

[2]OCAD University
Toronto
fchevalier@ocad.ca

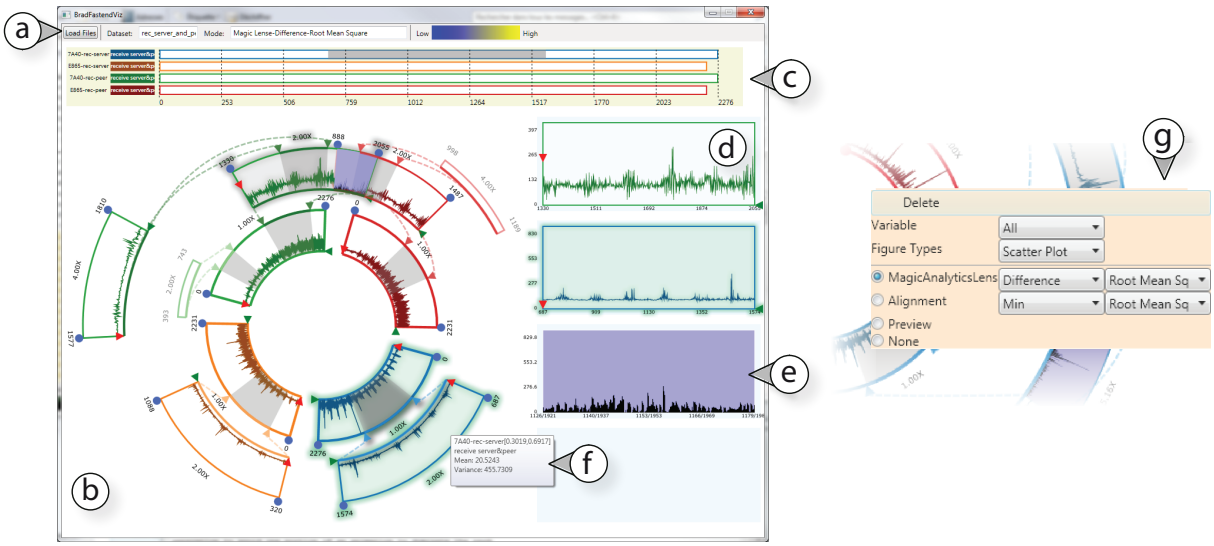


Figure 1. Exploration of four stock market datasets using KronoMiner. Left: the visualization window of KronoMiner including a (a) status bar, (b) main circular tree window (c) timeline overview, (d) four plots comparison area, displaying (e) a MagicAnalytics Lens detailed view. A tooltip (f) provides additional information on demand for each segment. Both general and context-based operations are available by invoking (g) a context menu.

ABSTRACT

The need for pattern discovery in long time-series data led researchers to develop interactive visualization tools and analytical algorithms for gaining insight into the data. Most of the literature on time-series data visualization either focus on a small number of tasks or a specific domain. We propose KronoMiner, a tool that embeds new interaction and visualization techniques as well as analytical capabilities for the visual exploration of time-series data. The interface's design has been iteratively refined based on feedback from expert users. Qualitative evaluation with an expert user not involved in the design process indicates that our prototype is promising for further research.

ACM Classification Keywords

H.5.2 User Interfaces: [Interaction Styles]

General Terms

Design, Human Factors

CHI 2011, May 7–12, 2011, Vancouver, BC, Canada.

INTRODUCTION

The need to analyze time-series data is prevalent in various application domains ranging from business planning to intensive care medicine. Interactive visualization has proven to be an effective approach to analyze time-series, assisting doctors in diagnosis [25], helping computer network engineers optimize bandwidth usage [22], aiding finance experts market investments [5] or helping homeland security investigations [6].

Existing interactive time-series visualization tools are either domain-specific or multi-purpose. Domain-specific tools are typically designed with domain experts and are rich in specialized functionality. They are powerful for answering precise questions (e.g. what is the impact of a medical treatment? [10]) but they cannot be easily applied to other domains. They also confine users into predefined tasks and procedures, thus offering little flexibility in the exploration process and leaving little room for opportunistic discoveries. In contrast, multi-purpose tools are not restricted to a specific domain and thus target a wider audience. However, they usually only support a small number of tasks (e.g. identifying periodicity [33]) and thus require other tools to be used in conjunction for complementary but necessary exploration capabilities.

There is therefore a need for interactive time-series data visualization tools that are functional enough to support complex tasks while being flexible enough for promoting opportunistic exploration in a domain-independent manner. Not only does domain-independence make the tool accessible to a wide audience, but it also allows cross-examination of data from multiple domains and opens up new possibilities for collaboration between analysts with complementary expertise.

To address this need we propose KronoMiner, a multipurpose time-series exploration tool providing rich navigation capabilities and analytical support. KronoMiner is based on a radial display that can be drilled into details by focusing on different pieces of the data and rearranging them in a facile manner. The design of KronoMiner has been refined based on expert users' feedback at several stages of the development. Our contributions include (i) a taxonomy of design requirements gathered from experts analyzing time-series from various application domains, (ii) a review of the design space of layouts for visualizing multiple multivariate time-series, (iii) the iterative design of KronoMiner as a demonstration of the new visualization and interaction techniques in a seamless tool, and (iv) expert user feedback on our prototype.

RELATED WORK

Visual Exploration is the process of interactively browsing through different portions of a dataset to gain a better understanding of it. Not only does Visual Exploration help for hypothesis confirmation, it is also crucial for finding the unexpected and for raising new questions, especially when the users have no or only vague hypotheses about the data [31]. Although time-series data is an area of extensive study in Information Visualization research, existing interactive visualization tools for time-series data do not extensively support multipurpose Visual Exploration: they are either too specialized domain-specific tools, answering precise questions, or domain-independent but limited to a few capabilities. Relevant surveys include [24, 29].

The prevalent approach for representing temporal data is arguably timeline visualization (e.g. LifeLines [25]) in which data points are usually plotted along a uniform time axis. Other approaches use visual metaphors to represent periodic temporal data. For instance, Calendar View [32] uses a calendar display to visualize univariate time-series aggregated on a daily, weekly or monthly basis to find interesting groupings of the data. SpiraClock [7] provides continuous feedback about upcoming events by displaying them on a spiral analog clock. A similar spiral shaped time axis is used in Spiral Graph [33] to reveal cycles of a time-series. All these techniques require the data to exhibit regularity, and therefore are of a little utility for data without periodicity.

For data without apparent cycles, multi-foci interaction is required for the detailed comparison of different pieces of the data while preserving context. For example, LiveRAC [22] and Mélange [9] rely on stretch and squish navigation. They result in a single integrated view with visual distortion, even though Richter et al. [26] suggest that non-linear representation of time has negative effects on the interpretation of the data. In contrast, StackZoom [17] preserves time axis uniformity for each single sequence of a multi-resolution hierarchy dynamically built by the user. However, the multiple foci views cannot be rearranged, thereby making it difficult to compare regions of interest that are spaced out.

Time-series visualization tools also frequently integrate analytical methods to support pattern mining. For instance PatternFinder [10] assists in detecting user-defined event patterns in medical records by letting the user specify the desired sequence of events, their attributes and inter-event time spans. Although pattern mining methods help to gain insight into data, most of them require prior knowledge of the domain and of what characterizes an interesting pattern. In the context of domain-independent exploration, no specific pattern mining methods can be easily applied. Nevertheless, combining visualization with general analytical methods such as the difference plot of two variables, remains useful as an indication for tentatively identifying regions of interest. In contrast to PatternFinder, TimeSearcher2 [5] relies on such a Visual Exploration process by allowing the user to dynamically select a timebox directly on the visualization as a query by example.

When they support pattern mining, time-series interactive visualization tools usually present relevant patterns as a list sorted by matching scores to a query [10, 5], or automatically align them [25]. Since they often focus on one pattern at a time, these tools fail in providing an overview of all the relationships within and across time-series. An effective way for showing the relationships is to explicitly link the regions that match to each others using straight and curved lines. However, this can quickly lead to visual clutter because of too many link crossings. A common approach to address this problem is to bundle the links [16] and represent the data on a circular layout, as it is done in MizBee [23]. A survey of circular visualizations can be found in [8].

To summarize, despite the abundant research for visualizing time-series data, as far as we know, there is no system that extensively supports domain-independent Visual Exploration of multiple multivariate datasets. Available interactive visualization tools are either highly domain-dependent and limited to predefined tasks and procedures, or general multipurpose tools that support a very few number of tasks. With the knowledge we gained through our literature survey, we derived the design guidelines listed in the next section we used as a basis for the design of KronoMiner.

DESIGN GUIDELINES

Before enumerating the design requirements KronoMiner has been built upon, we give background on time-series data and discuss the design space of layouts for visualizing them.

Time-Series Data

Time-series are sequences of data points measured at successive times, typically at equal intervals. They consist of real numbers (e.g. temperature logs), or events (e.g. medical histories made of records of doctor visits, blood tests, etc). Time-series have the common property of being large, not only in the captured time period and number of samples, but also in the number of observed attributes [2].

The characteristics of time-series vary widely and are very dependent on the application domain. The temporal dimension itself is a special dimension that has multiple facets. Aigner et al. [2] have listed the following criteria defining the different types of time: (1) structure of time (linear vs. cyclic), (2) temporal primitive (time points vs. time intervals) and (3) time perspective (ordered time vs. branching time). In this work, we focus on non-intrinsically-periodic linear sequences of ordered time points. Nevertheless, it is possible to generalize our framework to intervals.

One of the inherent limitations of traditional time-series interactive visualization tools lies in regarding time as being an immutable and uniform dimension. Depending on the tool, Visual Exploration is usually constrained to at least one of the following: (1) *synchronism*: different attributes are usually time-aligned, making it difficult to compare delayed sequences; (2) *chronology*: time-series data is often considered as ordered data, for which successive pieces cannot be rearranged; (3) *time scale consistency*: comparing time-series simultaneously at different time scales is rarely supported. For example, squishing or stretching the time scale of server logs independently makes it possible to adjust the visualization according to response-time capacity. Some users might also look for sub-patterns or fractal behaviour [12].

Design requirements

Before designing an interactive visualization system for Visual Exploration of time-series data, we conducted an informal study with experts from different application domains to understand what features users need and how they generally carry out their exploration of time-series data. Based on their comments and the knowledge we gained through our literature survey, we derived the following design requirements.

R1: Multivariate, multiple and heterogeneous datasets. Dealing with as varied input data as possible is important for supporting multipurpose use.

(a) *multiple heterogeneous datasets:* supporting time-series from various heterogeneous sources to facilitate across-concept comparison and refrain from being domain-dependent,

(b) *multivariate data:* visualization and manipulation of both coupled and dissociated attributes,

(c) *variety of chart representations:* providing different chart representations choices.

R2: Multiple coordinated views [2]. When user actions receive immediate visual feedback, multiple views aid Visual Exploration by providing diverse perspectives of the data.

(a) *large-scale overview:* acquiring general context at any exploration state through an overview of the entire datasets help maintain a mental map of the data [28],

(b) *multi-foci, temporal hierarchies:* showing several levels of detail simultaneously in coordinated views can aid in effective browsing of the data [17],

(c) *side-by-side comparison:* side-by-side comparison is easier than comparing visually spaced-out charts or remembering previously seen views [22].

R3: Interaction. Giving users as much freedom as possible to manipulate and query the data is crucial for Visual Exploration [2].

(a) *direct manipulation:* physically manipulating graphical representations of objects has the benefit of being easy to use and learn and allows for greater system comprehension [27],

(b) *overview first, zoom & filter, details on demand:* the widely followed Shneiderman's principle [28] is recognized as effective for dealing with large datasets,

(c) *dynamic multi-foci, temporal hierarchies:* drilling down into multiple regions of interest by time brushing (i.e. manually select data on an interactive data display) eases comparison [17],

(d) *grouping, aggregating for data abstraction:* grouping several regions of interest in a single entity allows for user-defined clustering and data abstraction,

(f) *history:* maintaining the history of the exploration process is important to help users revisit important views and keep track of their exploration [14],

(e) *annotating:* annotations aid for organizing and lower memory load as they act as labels and reminders.

R4: Analytical methods. All our expert users echoed that semi-guided exploration based on statistical analysis should be considered [2].

(a) *across-concept relationships:* displaying the relationships (links, scores) resulting from the computation of classical domain-independent analytical methods provides important clues for Visual Exploration,

(b) *best match:* finding the best match between series and their matching score allows for pattern mining [5],

(c) *dynamic comparison:* displaying on-the-fly computed information (e.g. difference plot) facilitates identifying regions of interest if immediate visual update is supported while manipulating the data,

(d) *customization:* allowing for user-defined operators avoids a multipurpose tool to be too limited for specialized domain-dependent analysis.

Design Space of Layouts

Here we review different layouts that can be used to represent time-series data as a complement to Meyer and Munzner's taxonomy [23]. Figure 2 depicts the possible layouts for multivariate (a,b) and multiple (c-f) time-series datasets. Note that an exhaustive taxonomy would include spiral layouts previously described, but these ones are suited for data that exhibits a periodicity and are out of the scope of this paper.

We distinguish between multivariate data, which implies the plots to be coupled, and multiple independent datasets (that could be multivariate or not). Multivariate data can be overlaid in a single plot (Fig. 2a), but visual clutter might be overwhelming; an alternative is to stack multiple charts (Fig. 2b). Different chart types can be set for each individual plot independently without overlap, but this requires more screen real estate and might split the user's visual attention.

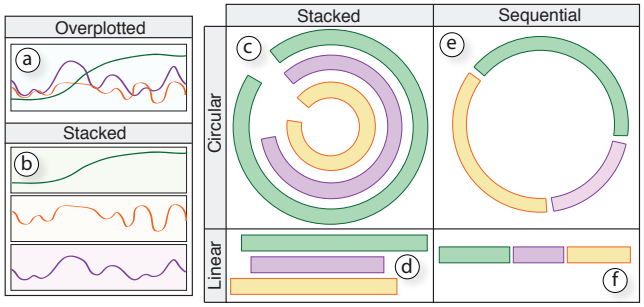


Figure 2. Design space of layouts for multiple multivariate time-series. a-b multivariate layouts, c-f multiple datasets layouts

For plotting more than one time-series dataset at a time, one can choose between a linear and a circular layout, and use stacked or sequential plots (Fig. 2). Linear layouts are more familiar to users and do not distort the data. Circular layouts are more appropriate for visualizing hierarchical structures and relationships [8]. They also successfully apply to periodic data when spirally-shaped, but this raises issues when considering multiple series with different periodicity and time interval as the series have to be piled up in the same spiral. The sequential linear view (Fig. 2f) makes it difficult to compare time-aligned data, especially when the plots are spaced out and cannot be rearranged. The circular stacked layout (Fig. 2c) preserves time alignment, but the same data plotted on different rings is perceived differently, and may cause misinterpretation. As described next, the KronoMiner interface combines both circular (Fig. 2e) and linear (Fig. 2d) layouts.

THE KRONOMINER INTERFACE

KronoMiner is an interactive visualization tool for Visual Exploration of multiple multivariate time-series datasets. Its design relies on the aforementioned guidelines and has been iteratively refined based on experts feedback gathered at different development stages (see User Feedback section). KronoMiner aims to ease the identification and manipulation of multiple sub-pieces of interest for finer analysis, rather than providing a single global view of numerous long time-series to be analyzed as a whole. It is based on dynamic hierarchical inspection of the data at different scales and locations. The interface (Fig. 1) combines several coordinated components, taking full advantage of the individual strengths of each layout to effectively support different tasks.

Multi-Foci Hierarchy Tree

Hierarchy Tree Visualization

The main interactive exploration window (Fig. 1b) displays a circular tree structure for presenting the hierarchy of multiple foci (R2b). We have chosen a circular sequential layout for the display of the multi-foci hierarchy mainly for its compact representation of tree structures [8], but also to put selectable data within easy reach of the user [11]. Each of the tree nodes corresponds to data points within a specific range of a parent series and is displayed as a concentric arc segment on which data is plotted. The central ring contains all the available entire datasets (R1a,b). Segments belonging to different datasets can be easily distinguished by the color of their outlines. Detailed information such as timestamp bounds and time scale is displayed for each segment.

A parent-child relationship of two segments is explicitly visualized as a pair of dashed curves that link one segment to the corresponding region of interest (ROI) on its parent. Data points belonging to the ROI are duplicated in the associated segment, as a new and more detailed view of the same initial dataset. To visually reinforce the hierarchical structure, ROIs are filled in with light gray, and arcs on both of the child and parent segments are color-coded to give awareness of which portion of this data piece resides in the series. A distinguishable hue is mapped to each dataset. The dashed links as well as the lower boundary of each child segment are colored with the same hue as its parent. The luminance conveys the position in the parent series (the brighter the sooner in the parent). Moreover, when hovering over a segment, its ancestors and descendants in the hierarchy and the associated ROI in the parent are emphasized by glow effects (Fig. 1). We also indicate its location in the timeline overview (R2a, Fig. 1b), and a tooltip showing detailed information is popped-up (R3b, Fig. 1f). Duplicated data is thereby clearly pointed out by providing immediate feedback on the different coordinated views as the user interacts (R2).

Building the Hierarchy Tree

As described in this section, the whole mechanism of creating and adjusting the segments relies on the direct manipulation principle (R3a) to support a quick and effective way of drilling down into multiple locations of the datasets (R3c). To enable quick learning of the interface, all the operations for editing the hierarchy are performed on the main interactive view, by simple mouse interaction and one modifier key.

A new segment is created by directly brushing (i.e. dynamic selection in the interactive view) the ROI on any existing series in the hierarchy (R3a,c). The user is required to right click on the desired starting point — a new segment appears on the ring located one level outside from the queried data piece, and the ROI is colored with light gray. The light gray highlight delimiting the ROI and its associated segment are dynamically adjusted while the user drags the cursor, until the mouse button is released at the intended ending point, finalizing the creation. When the user needs to focus on a specific time frame, she can perform time brushing on a collection of segments belonging to a ring at once. Simultaneous creation of as many child sequences as segments present on a ring can be done in a single operation by holding the *Shift* key while brushing the time frame on one of the ring's segments.

Figure 3 depicts how the multi-foci time intervals can be adjusted by direct manipulation on the view (R3a). A Ortho-Zoom mechanism [3] is used to adjust ROIs in the following way: the user can look at earlier or later data points in the parent series by dragging the ROI along the ring, keeping the same time frame but at different time locations (Fig. 3a). Dragging the cursor on the radial direction towards the center (Fig. 3b) reduces the interval. Conversely, a longer interval is obtained by dragging in the opposite direction. Arrows ending the dashed curves that link a segment to its corresponding ROI are also draggable and act as sliders to adjust the ROI size (Fig. 3c). The arc segment display can also be modified: round handles placed on the extremities of each

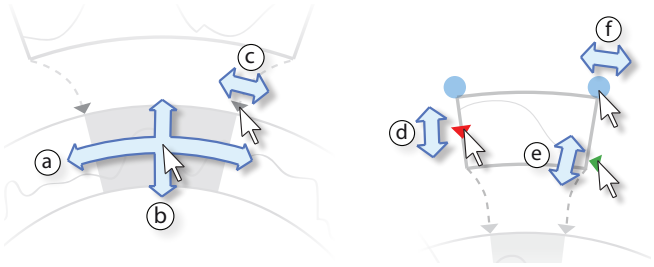


Figure 3. Left: adjusting a region of interest, (a) move and (b,c) resize. Right: adjusting a data arc segment, (d) change y-axis amplitude, (e) shif y-axis and (f) stretch or squish time

segment (Fig. 3f) allow for stretching or squishing the data representation, thus affecting the time scale used for rendering (the data always occupies all the available segment space), the ROI and the data itself remain the same. The user can also set up specific values by invoking the segment setting pop-up menu as she clicks on the segment's time scale text label on its top (Fig. 1g), both the related ROI and segment are updated in consequence. Two more controllers (Fig. 3d,e) explained later, are provided to the user for data plot visual settings.

Editing the Hierarchy Tree

Maintaining the history of the exploration process (R3f) is important to help users revisit important views and keep track of their exploration [14]. Even though we do not explicitly support navigation history, the hierarchy tree mimics exploration history since a new segment is created each time the user wants to drill down into the data and all the existing segments remain in place. When the user wants to focus on the deepest levels of the hierarchy because they correspond to the finest details, she does not require the whole hierarchy tree to be displayed in detail. In the circular representation, the deepest levels are located on the outer rings. To release space and focus visual attention, the user can shrink the inner rings (see Fig. 4) by dragging a ring towards the center of the circle view — ring interaction (Fig. 5c) is detailed later.

To avoid the view being too crowded because of the presence of too many segments, the user can also put data aside while keeping it at her disposal for later. Double clicking on a segment makes it collapse: a segment of a reduced height, not displaying any details then takes its place. Hence visual clutter is reduced, making it possible to focus on a small subset of visually detailed segments without losing the mental map of the current structure. The details can be restored anytime by a double click on a collapsed segment. Figure 1 shows examples of both collapsed and detailed segments.

Keeping the whole history of exploration is sometimes not relevant when intermediate stages are not important for the analysis. For example when a sequence of interest has been identified, the user might no longer be interested in its ancestors in the hierarchy. To get rid of unused segments, the user can enter the *Deletion mode* (discussed later) and delete either a single segment by clicking on it, or a segment and its whole sub-hierarchy while holding the *Shift* key. The deleted pieces disappear in a smooth fade out, avoiding abrupt changes in the view when deleting a whole branch.

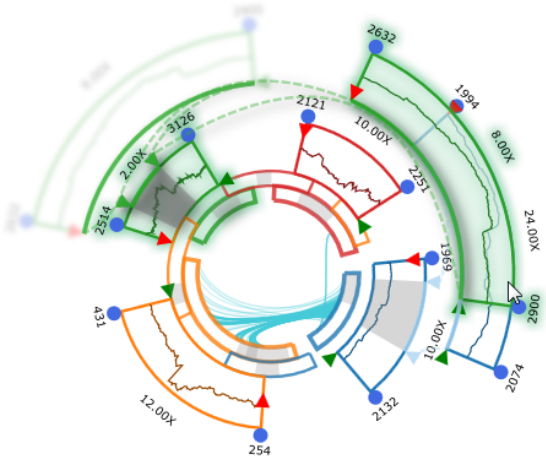


Figure 4. Example of exploration of the NYSE exchange daily open price (1970-2010) using the *Preview mode*: a phantom of the manipulated segment remains at the initial position (top left) as it is temporary overlaid on another segment for trend comparison. The inner rings have been shrunk to focus on the deepest levels of the hierarchy, and the relationships between the datasets are shown in the center.

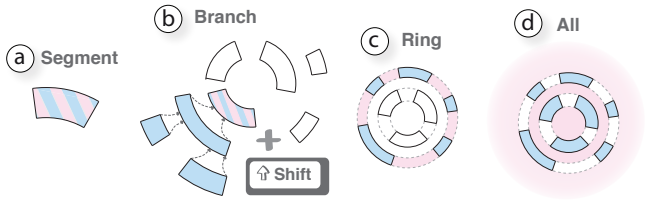
Flexible Interaction for Visual Exploration

Adjusting the Layout for Segment Comparison

The user can freely adjust the main interactive view through classical pan (right button drag), zoom (mouse wheel) and rotate (left button drag) operations to access details. When looking at time-series data, one of the main tasks is to compare data at different locations in time. However, two segments can be spaced out in the view, even showed upside down because of the circular display. This can make the comparison more difficult than looking at side-by-side pieces (R2c). To facilitate comparison, we give the user the full freedom of rearranging the segments in the view: a segment can be rotated anywhere in its own ring. Ring jumps are also supported. Thus the segments can be freely aligned either on a stacked configuration by putting them on different rings, or overlaid to obtain overplotted data (Fig. 2a) as they are semi-transparent. We discuss later that the overlapped area can be used for analytical purposes by displaying the result of statistical functions instead of the raw data.

Looking for the unexpected is difficult and tedious and usually requires going through a lot of operations before serendipitously discovering relevant insights within the data. To avoid the user messing up her layout during the Visual Exploration, we allow for a *Preview mode* in which the current layout is maintained while the user performs a temporary analysis (e.g. dragging a segment on top of another). A phantom of the segment remains in place at its initial position, and is restored as she releases the mouse button. Figure 4 shows a temporary comparison of similar trends of open price at different time scale and location by overlaying one segment over the other in the *Preview mode*.

KronoMiner is designed to be as unconstraining as possible in terms of comparing multiple sub-pieces of several datasets. Much freedom is given to the user: segments from any loaded dataset, at any location in the dataset and rendered at any time scale can be placed anywhere in the view. Easy access



		2	Drag	Drag	Wheel
a	Select	Collapse Extend	Rotate Ring jump	Create and adjust a new segment	
b					
c				Shrink inner rings	
d			Rotate	Pan	Zoom

Figure 5. Context-based interaction. The activation area is showed in pink, and the affected elements in blue. Interaction can be performed on (a) a single segment, (b) a segment and its children branch, (c) a whole ring and (d) all rings at the same time.

to the operations through the direct manipulation techniques (R3) allows for a quick and smooth manipulation of the data with immediate visual feedback. However, giving too much freedom to the user also has its drawbacks. The hierarchy structure might not always be easy to read after reorganizing the segments. Visual cues have to be integrated to help the user maintain her mental map. To this end, we use explicit dashed links, color coding and immediate highlighting of the related branching (ancestors and children in the hierarchy) while hovering over a segment. All these techniques are designed to give as many visual cues as possible to keep awareness of the overall tree structure.

Context-based Interaction

We introduce a new context-based interaction language for the user to be able to manipulate more than one segment at a time. We exploit the current position of the cursor to determine whether the user intended to perform an action on a) a single segment, b) a segment and its whole branch, c) a single ring or d) the whole view. Figure 5 shows a summary of the different selectable groups, and the available operations for each group using the mouse buttons and the *Shift* modifier key. The pink areas correspond to all the cursor positions that determine a group, which is depicted in blue. For example, by holding the *Shift* key and dragging a segment, the user may performs a "branch" move operation (Fig. 5b) and then can rotate a segment and all its descendants in the same interaction or make them jump to different rings. Context-based interaction allows the user to adopt different strategies for Visual Exploration and optimize the interaction. For example, the rings can be used to categorize the data; each ring being dedicated to a specific group the user wants to keep related (R3d). By performing actions in the ring-mode, the user can apply the same changes to all the elements of the group in a single interaction while keeping consistency within the group.

Keeping consistency while designing direct manipulation is possible although limited: there is a finite number of ways of binding operations to the combination of mouse buttons, mouse wheel and modifier keys. Other interaction techniques such as context menu or hotbox [21] must be integrated as a complement to direct manipulation to offer more complex operations and capabilities. Following the same context-based principle just described (Fig. 5), we make available additional functionality (discussed later) on a context menu that is invoked by hitting space bar (Fig. 1e). The user can perform deletion operations and visual settings that affect a targeted set of segments, depending on the position where the menu has been invoked. The menu also contains permanent options, such as the different modes available, among which the *Preview mode* and the *Deletion mode* we introduced earlier.

The Detailed Comparison Window

Circular layouts suffer from distorting the data and might be constraining for a detailed analysis. To balance the drawbacks of the radial view, we have designed a dedicated *Detailed Comparison Window*, located on the right of the interface (Fig. 1c) where four selected segments can be displayed at a time, in a traditional linear stacked layout. To display a segment in the comparison window, the user is required to click on the desired targeted space among the four available, then click on the desired segment to be displayed. If a segment is already plotted, the new selection replaces it. The user can also right click on any of the four linear plots to clear it.

The Detailed Comparison Window was initially designed as two side-by-side plots. The current active plot was permanently showed, the second slot being reserved for the segment under the mouse cursor. One of our experts involved in the iterative design process said he *"[wanted] to be able to look at more than two plots at a time"* and that *"they should be constantly visible after [he had] selected them"*. He also reported that *"seeing them displayed on top of each others would facilitate comparison as I need them to be aligned"*. We currently support four plots, but one can easily imagine more to be visualized by making scrollable the detailed view.

To facilitate data browsing, the comparison window is enriched by interaction capabilities: the user can pan the data in the linear plot for further exploration or to control time alignment. The corresponding ROI and its associated segment are updated in the meantime. Two controllers (similar to Fig. 3d,e) are available on the duplicated linear segments, to set up y-axis amplitude and shift. We guarantee a full coordination between the two views so that the user can easily keep track of the impact of one operation on the other view. Moreover, both the arc segment and its copy are highlighted while hovering over one or the other to make it clear which data is common to the coordinated views (R2).

Dynamic Analytical Methods

When analyzing time-series data, users are commonly interested in the evolution of their data over time for detecting trends and patterns, but also anomalies. Analytical methods can help gain insight into data by automatically detecting periodicity, recurrent patterns and self-similarity [2]. However, one of the major problems is that most of these methods are heavily domain-dependent and thus cannot be applied to any type of data, and even less for comparing time-series from different application domains. Moreover, the parameters are often not self-explanatory and hence not easy to set, especially without some prior knowledge of the datasets and the characteristics of patterns of interest.

When the user has no or a vague idea of the patterns she is looking for, precise analytical computations answering specific questions are of little use. In this case, the user has to rely on Visual Exploration in order to find out the questions she does not know yet. Basic, general and domain-independent analytical computations (e.g. difference plot) are then crucial as they provide clues and suggestions of regions worth exploring deeper. Tools such as TimeSearcher2 [5] that allow for query by example pattern finding, have already demonstrated the benefit of integrating interactive visualization and analytical capabilities in a single tool. KronoMiner enriches its Visual Exploration capabilities by integrating analytical analysis support in three different ways: (1) visualization of the relationships within and between the datasets, (2) the MagicAnalytics Lens: on-the-fly computation and display of analytical methods comparing two segments, and (3) the Best Match mode: on-the-fly computation and display of the best match of two segments. We detail them in the next sections.

Showing the Relationships

Although a lot of existing interactive visualizations allow for pattern mining, to our knowledge, they all only focus on a single pattern at a time. As a consequence, they fail in providing the user with an overview of all the relationships existing within and between the datasets. In other application domains, such as Biology, it is common to represent the entire set of links representing relationships (e.g. MizBee [23] displays matching subsequences of two chromosomes). Showing all the relationships at once help the user gain a overall idea of both the closely related regions and isolated data, that are usually starting points for exploration.

In KronoMiner, we compute similar patterns using a simple brute-force similarity mining algorithm at the loading of the datasets. Any other algorithm or even predefined links can be used. Similarly to MizBee [23], the relationships within and between the different entire datasets are displayed at the center of the circular layout, connecting the data items of the main ring. The links intensity is mapped to the matching score (the darker, the higher). To make trends visually obvious we combine the edge bundling technique described in [16] with alpha blending resulting from overlapping semi-transparent links. In this way, the user has a constant access to relationships information she can use for visually mining regions of interest (R4a), thereby guiding Visual Exploration.

MagicAnalytics Lens

Detailed pairwise comparison is one of the main tasks when analyzing time-series data. It usually involves computing the correlation between the two series to compare (or subsequences of them), sometimes at a different time scale; and viewing the similarity scores of discovered patterns. Traditional statistical tools, such as Matlab, allow for a fine analysis of the data: a wide variety of predefined analytical methods are available, as well as the possibility of defining new functions. However, in such tools the user is required to specify the exact instructions (e.g. the bounds of the interval to focus on) through the use of a command line before visualizing the result. In the case that she does not know exactly what to look for, she might generate a lot of plots before finding something relevant. Not only is the process repetitive, haphazard and time-consuming, it is also distracting as the user splits her attention focusing back and forth between the command interface and the visualizations, thereby requiring she remembers what has already been explored. Moreover, these tools usually require some basic programming skills, which narrows the target audience.

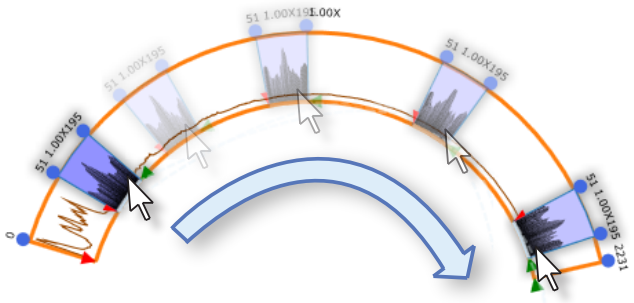


Figure 6. Example of use of the *MagicAnalytics Lens* on peer-to-peer activity logs. A segment that roughly corresponds to one day of activity is used as a lens. The lens displays the correlation plot, and the background color conveys the global root mean squares operator on a yellow-to-blue scale, in this case showing that a daily pattern is repeated as the user drags the lens along the series.

To avoid breaking the flow of the exploration process and facilitate opportunistic discovery, we introduce the *MagicAnalytics Lens*, an interactive visualization technique that computes in real time the result of a function involving two time-series, and immediately displays its result (R4c). MagicAnalytics Lens is inspired by Magic Lens [4] that affects the image inside the bounds of a shape defining the lens. In contrast to Magic Lens that uses a dedicated window as a lens applying a fixed predefined operator, MagicAnalytics Lens considers all the data pieces as a potential lens, as the displayed result depends on both the data points in the window used as a lens, and the data points under it. In Figure 6 the lens shows the cross-correlation plot between two series using the data contained in the overlapped interval. The plot is automatically updated while changing the alignment as the user moves or stretches the segment used as a lens. The background color of the lens is mapped to a computed global measure (root mean square in Fig. 6). This way, the user can very quickly assess if the current alignment compares related series. Any segment in the view can be used as a MagicAnalytics Lens, making it possible to compare data both within the original series and between the different datasets, and also data at different time scales as the user can stretch and squish the segments. A MagicAnalytics Lens can also be plotted in the synchronized Detailed Comparison Window to access the details as showed in Fig. 1c.

The MagicAnalytics Lens is made available through the contextual menu (Fig. 1g). When entering the mode, all the overlaid areas are replaced by the result of the selected functions. The current prototype integrates general functions such as the difference plot, the minimum and maximum plots, and the inner product or the root mean squares for the background, but any operator can be embedded. Ideally, the user should be able to define her own functions and then customize the tool by introducing new analysis methods on demand (R4d).

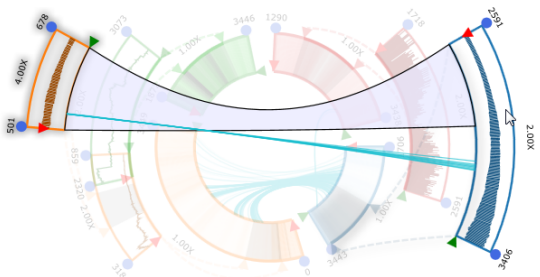


Figure 7. Example of Best Match between two segments of NYSE exchange daily open price (1970-2010) minimizing the root means square. An arch conveys the best match location, its background color is mapped to the matching scores on a yellow-to-blue scale. The pre-computed links are replicated between the two sequences. In this example, the relationships between the segments surprisingly do not reflect the best match location and might be worth exploring further.

Finding the Best Match

There is an abundant literature about time-series statistical tools that are dedicated to pattern retrieval. Pattern discovery is an important task while analyzing time-series data as it is the preliminary step to discover new ways of defining what could characterize similarity. Allowing users to select a pattern from the data itself has been showed to be useful [5], especially in the context of Visual Exploration when the user does not have a precise idea of what she is looking for.

KronoMiner integrates a *Best Match* mode (R4b): the current selected segment defines the query pattern and hovering over a segment defines it as being the target. The best match according to a selected similarity measure is dynamically computed and displayed: an arch links the query segment to its best match within the targeted segment and the rest of the main view is faded out to avoid visual clutter (Fig. 7). The background color of the arch shape is mapped to the matching score to provide a quick indication of the similarity. We also replicate the same pre-computed relationships as showed in the center of the radial display. We thus make available the context that would otherwise be lost because of fading out. The user can access the details anytime by hovering over the arch if she needs to know the precise score value and detailed information about the two matched segments.

Similarly to the MagicAnalytics Lens, the Best Match takes into account the current time scale of the segments, thus allowing for comparing sequences at a different time scale. For example, fractal patterns within the same time series could be revealed if the user detects a high similarity between a segment and the series it comes from. The similarity function to be used is selectable through the context menu (Fig. 1g), but any other pattern mining method could be integrated.

Plot Visual Settings

For a detailed analysis, the user is able to change the plot style among the six different available (Fig. 8) in the context menu (Fig. 1g) — other types such as horizon graphs [13] or heatmaps could easily be added. Note that the amplitude and bubble plots particularly suit for radial layout as their perception is less affected by the circular deformation. We also support data display adjustments by allowing the user to alter the y-axis amplitude and shift. To do so, two draggable handles are made available on each side of a segment in both the circular main view and the comparison window (Fig. 3d,e). Dragging up and down the left handle respectively increases and decreases the amplitude. The right handle determines the position of the origin of the y axis. Dragging it up and down shifts the chart display. All the rendering settings in one of the two views are automatically reflected in the other, guarantying coordination and consistency (R2).

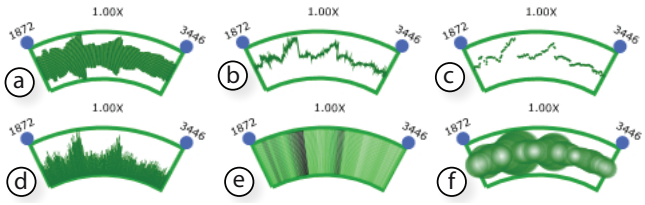


Figure 8. The different available plot styles: (a) amplitude plot, (b) line chart, (c) scatter plot, (d) histogram, (e) lines plot and (f) bubble plot.

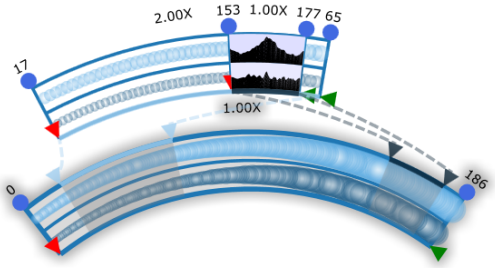


Figure 9. Example of use of the MagicAnalytics Lens on a multivariate dataset. Top plot: monthly average cost of a night's accommodation in Victoria, bottom: Consumer Price Index in Melbourne (1980-1995). The correlation plot and the background color (root means square) indicate common trends between the two segments.

One might have to deal with multiple variables while keeping them coupled. Examples include coordinates of a moving object, min/max temperatures and opening/closing price of stock market that would not make sense desynchronizing. We support multivariate datasets by piling up multiple views as showed in Figure 9. In this example, the MagicAnalytics Lens reveals a repeated pattern, but the trend is observed at a different time scale. A room price and the CPI evolve in a similar way at different times, but twice as fast the second time. Experts could make the hypothesis that there exists a correlation, and then potentially predict the phenomena when the pattern occurs again.

Although it is possible to make rings larger to fit more than 3 variables at a time, KronoMiner does not scale to larger dimensions because of visual clutter. Design improvements (e.g. dynamic reordering and stacking) are needed to support high dimensional data. Space-efficient alternate visualizations (e.g. braided graphs [18]) would partially solve the problem but imply issues for the analytics lens design.

USER FEEDBACK

The KronoMiner interface has been iteratively refined through 4 participatory design sessions with 7 experts from different domains at different stages of the development. Each session focused respectively on Computer Network (1 expert), Medical Science (1), Weather Forecast (3) and Finances (2) and lasted 4 to 5 hours during which we introduced the tool, let the users try it and gathered feedback and requirements.

Moreover, an expert user not involved in the design process has tested the last prototype on his own data in 2 working sessions of 3 hours each. The participant is a researcher studying large scale peer-to-peer traffic data on video-on-demand websites, and usually uses Matlab. After describing the tool interface and functionalities, the user was asked to explore his data using the think-aloud protocol.

We first observed the user was clearly more comfortable with the interface during the second interview, whereas he complained *"it is difficult to remember, I have to learn"* when testing the earlier prototype. At the end of the sessions, he compared the learning process: *"It took me weeks, even months to learn how to use Matlab because I had first to learn code. I can do a lot with your tool, and I don't even have to think about it"*. We strived to make the interaction as direct as possible so as not to break the exploration flow. Context-based interaction requires training, but one cannot benefit from complex actions while always keeping straightforward interactions. We believe a novice user to be able to quickly get familiar with the different functionalities but this remains to be validated in further studies.

Our expert's dataset consisted of a month of the visiting population of two different channels, sampled from the server every 10 minutes. He first wanted to test the hypothesis that the sub-sequences of everyday's data are similar within and between channels. To do so, he brushed a ROI on one channel corresponding to what he described as *"the first day, roughly"* (he was able to identify it because of the characteristic shape of the plot). To refine his selection, he displayed the segment in the comparison window, changed the plot type to gain a better view and barely adjusted the time frame, pointing that *"it doesn't have to fit exactly, I just want a quick look"*.

The user panned the plot in the comparison window to get an overall idea of day-to-day data and possibly spot anomalies or interesting behavior he wouldn't see in the radial representation. He adjusted the scale and the shift of the y-axis as the amplitude of the plot decreased. At some point, he mentioned *"the plot is too small now, I can't say if I see a pattern, or if it is noise. So far I've been able to barely identify a similar pattern, but now it's not possible anymore, and increasing the amplitude wouldn't help"*. He thus decided to switch to the MagicAnalytical Lens mode as a support to analyze the cross correlation. As he dragged the *"first day"* segment along the sequence, he observed a constant peak on the center of the computed plot (Fig. 6), indicating a high correlation at a zero delay. This justified part of his hypothesis and he defined the lens dragging as being "*very handy"*.

The participant roughly selected the *"last day"* data points and dragged both days segments to outer rings to gain a better view, and made them overlap, still in the MagicAnalytics Lens mode. He then dragged the first segment to sequentially compare the data from the latest day to the rest of the sequence, and validated his hypotheses as they revealed being correlated. While doing so, our user emphasized that *"In Matlab you have to generate tons of graphs. Even if you have enough space to display them, you don't have enough brain. Here, all you have to do is to simply drag [the segment]"*.

The user started to explore the data on the secondary channel. He found the daily cross correlation to be high between the two datasets, but not as strong as within the same series. After extensive navigation and digging into the data, the work space turned to be messy. Our user started to rearrange the segments: "*I need to organize*", shrunk inner rings, and collapsed some remaining segments, as they were not of interest at this time *"I don't need this guy for now..."*. When cleaning his space, he realized he looked at many plots without noticing: "*This saves me a lot of time. Normally I have to eyeball all the data first because I don't want to go into every pieces because it is tedious"*. Further, he gathered unexpected findings as he dynamically adjusted the range of the ROI to time intervals less than a day - *"The data around midnight is similar to that around noon! It's interesting. I would have expected it is the case at other rush hours, but people should sleep at midnight. It is worthwhile to study deeper"*.

Our user was enthusiastic about analytical capabilities within a tool supporting direct manipulation. He said he wanted to explore more datasets with more variables and longer time periods using our prototype, as it would be too time-consuming with traditional tools. He mentioned that for analyzing server data, he would require to freely "*stretch the time*" as servers likely have different hardware. Hence the performance plots have to be adjusted for a fair comparison. The Best Match Mode would also be helpful as the need for mining guidance increases with the number and size of the time-series.

DISCUSSION

Look: sexy or serious?

When they do not propose a sober interface that exhibits several menus, control panels and numerous settings, tools for analysis are not taken seriously as they do not look professional. It is often that these tools are admitted to be nice-looking and pleasant to "play" with, but their appearance has a negative impact on the users' reliance. A tool that feels too easy to use might suffer from its apparent simplicity. Paradoxically, users are highly demanding for more interactivity and more user-friendly interfaces and thus create a tension between professionalism and accessibility. KronoMiner demonstrates that it remains possible to perform effective analysis while offering high interactivity in a seamless tool. There is a strong need to convince users to accept that the absence of menus and settings panels does not mean a tool is not powerful. Our user who was baffled at the beginning *"I've never seen that before"*, ended up asking us for a release of the prototype to use it for his research.

Back to the Roots

The previous observations raise the question of breaking habits. Integrating familiar interaction, and familiar representations seems to be essential for the users to feel confident and consider the tool potentially useful. Surprisingly, our expert has not been able to start any constructive exploration during our first meeting as he *"[couldn't] work without seeing somewhere some plots stacked in a linear view"*. However, he did not use the Detailed Comparison Window as much as expected during the second interview. To his own surprise, he admitted that it was reassuring to be able to get

back to the traditional representation, *"I know it is there, so I feel comfortable using the tool now"*. It is difficult to make an interface feel familiar since people have different working habits, especially when targeting a wide audience. We strived to design KronoMiner to be as flexible as possible for multipurpose analysis through participatory design studies deliberately involving researchers from disparate domains, as a guarantee not to specialize to a domain and an attempt to cover common needs. By doing so, we think our tool is adapted to a wide audience. In return, it is still too general to fully support deep analysis, as our user quickly missed analytical operations he is used to apply - *"It would be great if you can add more operators to the lens"*. Allowing users to customize the tool has revealed to be crucial for functionalities extension purpose. We also believe that this would help them accept more easily new ways of interacting, as they could personalize the tool to fit with their needs.

CONCLUSION AND FUTURE WORK

This paper introduces KronoMiner, a multipurpose tool for exploring time-series datasets. KronoMiner integrates new interaction and visualization techniques in a seamless tool designed to be as unconstraining as possible so as to ease the exploration of multiple sub-pieces of interest for finer analysis. Although it does not solve all the problems, KronoMiner illustrates how multi-foci navigation enriched with general statistical tools can help users perform exploratory analysis on multiple time-series data. Feedback from experts confirmed our belief that users benefit from systems combining both improved algorithms and interactive visual interfaces to identify trends or spot anomalies. In particular, the MagicAnalytics Lens that provides an immediate visual feedback has revealed to be a promising technique.

Despite the positive early feedback, longitudinal studies still have to be run to validate our claims and better spot the limitations. We plan to provide all our experts with a release of the tool for them to try it deeper on their own data and gather their feedback. We will also encompass a mechanism for the user to be able to customize the analytical tools to study how far a general tool can support tasks the users are used to perform with their traditional specialized systems.

REFERENCES

1. W. Aigner, S. Miksch, W. Müller, H. Schumann, and C. Tominski. Visualizing time-oriented data-a systematic view. *Comput. Graph.*, 31(3):401–409, June 2007.
2. W. Aigner, S. Miksch, W. Müller, H. Schumann, and C. Tominski. Visual methods for analyzing time-oriented data. *IEEE TVCG*, 14(1):47–60, 2008.
3. C. Appert and J.-D. Fekete. Orthozoom scroller: 1d multi-scale navigation. In *Proc. CHI'06*, 21–30, 2006.
4. E. A. Bier, M. C. Stone, K. Pier, W. Buxton, and T. D. DeRose. Toolglass and magic lenses: the see-through interface. In *SIGGRAPH '93*, 73–80, 1993.
5. P. Buono, A. Aris, C. Plaisant, A. Khella, and B. Shneiderman. Interactive pattern search in time series. In *Proc. of Visualization and Data Analysis*, 175–186, 2005.
6. W. Chung, H. Chen, L. G. Chaboya, C. D. O'Toole, and H. Atabakhsh. Evaluating event visualization: a usability study of coplink spatio-temporal visualizer. *Int. J. Hum.-Comput. Stud.*, 62(1):127–157, 2005.
7. P. Dragicevic and S. Huot. Spiraclock: a continuous and non-intrusive display for upcoming events. In *Proc. CHI'02*, 604–605, 2002.
8. G. M. Draper, Y. Livnat, and R. F. Riesenfeld. A survey of radial methods for information visualization. *IEEE TVCG*, 15(5):759–776, 2009.
9. N. Elmqvist, Y. Riche, N. H. Riche, and J.-D. Fekete. Melange: Space folding for visual exploration. *IEEE TVCG*, 16(3):468–483, 2010.
10. J. Fails, A. Karlson, L. Shahamat, and B. Shneiderman. A visual interface for multivariate temporal data: Finding patterns of events across multiple histories. *Symposium On Visual Analytics Science And Technology*, 0:167–174, 2006.
11. P. Fitts and J. Peterson. Information capacity of discrete motor responses. *Jrnl of Experimental Psychology*, 67:103–112, 1964.
12. J. Gao, Y. Cao, W.-w. Tung, and J. Hu. *Multiscale Analysis of Complex Time Series: Integration of Chaos and Random Fractal Theory, and Beyond*. Wiley-Interscience, 2007.
13. J. Heer, N. Kong, and M. Agrawala. Sizing the horizon: The effects of chart size and layering on the graphical perception of time series visualizations. In *Proc. CHI'09*, 2009.
14. J. Heer, J. Mackinlay, C. Stolte, and M. Agrawala. Graphical histories for visualization: Supporting analysis, communication, and evaluation. *IEEE TVCG*, 14:1189–1196, 2008.
15. H. Hochheiser and B. Shneiderman. Dynamic query tools for time series data sets: timebox widgets for interactive exploration. *Information Visualization*, 3(1):1–18, 2004.
16. D. Holten. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE Transactions on Visualization and Computer Graphics*, 12:741–748, 2006.
17. W. Javed and N. Elmqvist. Stack zooming for multi-focus interaction in time-series data visualization. In *PacificVis'10*, 33–40, 2010.
18. W. Javed, B. McDonnel, and N. Elmqvist. Graphical perception of multiple time series. *IEEE TVCG*, 16(6):927–934, 2010.
19. E. J. Keogh and M. J. Pazzani. Relevance feedback retrieval of time series data. In *SIGIR'99*, 183–190, 1999.
20. R. Lopez-hernandez, D. Guilmaine, M. J. Mcguffin, and L. Barford. A layer-oriented interface for visualizing time-series data from oscilloscopes. In *PacificVis'10*, 41–48, 2010.
21. M. J. McGuffin and I. Jurisica. Interaction techniques for selecting and manipulating subgraphs in network visualizations. *IEEE TVCG*, 15(6):937–944, 2009.
22. P. McLachlan, T. Munzner, E. Koutsofios, and S. North. Liverac: interactive visual exploration of system management time-series data. In *Proc. CHI'08*, 1483–1492, 2008.
23. M. D. Meyer, T. Munzner, and H. Pfister. Mizbee: A multiscale synteny browser. *IEEE TVCG*, 15(6):897–904, 2009.
24. W. Mller and H. Schumann. Visualization methods for time-dependant data-an overview. In *Proc. of the 2003 Winter Simulation Conference*, 737–745, 2003.
25. C. Plaisant, B. Milash, A. Rose, S. Widoff, and B. Shneiderman. Lifelines: visualizing personal histories. In *Proc. CHI'96*, 221–227, 1996.
26. H. A. Richter, J. A. Brotherton, G. Abowd, and K. N. Truong. A multi-scale timeline slider for stream visualization and control. Technical report, Georgia Institute of Technology, 1999.
27. B. Shneiderman. Direct manipulation: A step beyond programming languages. *Computer*, 16:57–69, 1983.
28. B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *VL*, 336–343, 1996.
29. S. F. Silva and T. Catarci. Visualization of linear time-oriented data: A survey. *Proc. of the Int. Conf. on Web Information Systems Engineering*, 310, 2000.
30. C. Tominski, J. Abello, and H. Schumann. Axes-based visualizations with radial layouts. In *SAC'04: Proc. of the Symp. on Applied computing*, 1242–1247, 2004.
31. J. W. Tukey. *Exploratory Data Analysis*. Addison-Wesley, 1977.
32. J. J. Van Wijk. Cluster and calendar based visualization of time series data. In *Proc. INFOVIS'99*, 4–9, 1999.
33. M. Weber, M. Alexa, and W. Müller. Visualizing time-series on spirals. In *Proc. INFOVIS'01*, 7, 2001.